International Journal of Latest Research in Engineering and Science
Volume 1 | Issue 3 | July 2015

Performance and Cost Evaluation of an Adaptive Encryption Architecture for Cloud Database

Ramana Reddy[1] M.Premchander [2]

[1]*PG Scholar, Department of Computer Science and Engineering, Maheshwara Engineering College, Hyderabad*

[2] *Head of the Department, Department of Computer Science and Engineering, Maheshwara Engineering College, Hyderabad*

Article History

Received on: 27-06-2015
Accepted on: 01-07-2015
Published on: 05-07-2015

Keyword
Cloud database
Power Quality,
encryption,
adaptivity,
cost model.

ABSTRACT

The cloud database as a service is a novel paradigm that can support several Internet-based applications, but its adoption requires the solution of information confidentiality problems. We propose a novel architecture for adaptive encryption of public cloud databases that offers an interesting alternative to the tradeoff between the required data confidentiality level and the flexibility of the cloud database structures at design time. We demonstrate the feasibility and performance of the proposed solution through a software prototype. Moreover, we propose an original cost model that is oriented to the evaluation of cloud database services in plain and encrypted instances and that takes into account the variability of cloud prices and tenant workloads during a medium-term period.

Corresponding Author

Ramana Reddy
Department of CSE
Maheshwara Engineering College
Hyderabad

I. Introduction

The cloud computing paradigm is successfully converging as the fifth utility [1], but this positive trend is partially limited by concerns about information confidentiality [2] and unclear costs over a medium-long term [3], [4]. We are interested in the database as a service paradigm (DBaaS) [5] that poses several research challenges in terms of security and cost evaluation from a tenant's point of view. Most results concerning encryption for cloud-based services [6], [7] are inapplicable to the database paradigm. Other encryption schemes that allow the execution of SQL operations over encrypted data either have performance limits [8] or require the choice of which encryption scheme must be adopted for each database column and SQL operation [9]. These latter proposals are fine when the set of queries can be statically determined at design time, while we are interested in other common scenarios where the workload may change after the database design. In this paper, we propose a novel architecture for adaptive encryption of public cloud databases that offers a proxy-free alternative to the system described in [10]. The proposed architecture guarantees in an adaptive way the best level of data confidentiality for any database workload, even when the set of SQL queries dynamically changes.

II. Requirements Elicitation

A. Existing System

Most results concerning encryption for cloud-based services are inapplicable to the database paradigm. Other encryption schemes that allow the execution of SQL operations over encrypted data either have performance limits or require the choice of which encryption scheme must be adopted for each database column and SQL operation. These latter proposals are fine when the set of queries can be statically determined at design time, while we are interested in other common scenarios where the workload may change after the database design.

B. Proposed System

The proposed architecture guarantees in an adaptive way the best level of data confidentiality for any database workload, even when the set of SQL queries dynamically changes. The adaptive encryption scheme, which was initially proposed for applications not referring to the cloud, encrypts each plain column to multiple encrypted columns, and each value is encapsulated in different layers of encryption, so that the outer layers guarantee higher confidentiality but support fewer computation capabilities with respect to the inner layers. The outer layers are dynamically adapted at runtime when new SQL operations are added to the workload. Although this adaptive encryption architecture is attractive because it does not require to define at design time which database operations are allowed on each column, it poses novel issues in terms of applicability to a cloud context, and doubts about storage and network costs. We investigate each of these issues and we reach three original conclusions in terms of prototype implementation, performance evaluation, and cost evaluation. We initially design the first proxy-free architecture for adaptive encryption of cloud databases that does not limit the availability, elasticity and scalability of a plain cloud database because multiple clients can issue concurrent operations without passing through some centralized component as in alternative architectures. Then, we evaluate the performance of encrypted database services by assuming the standard TPC-C benchmark as the workload and by considering different network latencies. Thanks to this test bed, we show that most performance overheads of adaptively encrypted cloud databases are masked by network latencies that are typical of a geographically distributed cloud scenario.

Finally, we propose the first analytical cost estimation model for evaluating cloud database costs in plaintext and encrypted configurations from a tenant's point of view over a medium-term period. This model also considers the variability of cloud prices and of the database workload during the evaluation period, and allows a tenant to observe how adaptive encryption influences the costs related to storage and network usage of a database service. By applying the model to several cloud provider offers and related prices, the tenant can choose the best compromise between the data confidentiality level and consequent costs in his period of interest

III. Literature Survey

A. Compostable cost estimation and monitoring for computational applications in cloud computing environments.

With the from cloud computing providers, scientists have the opportunity to utilize pay-as-you-go resources together with their own and shared resources. However, scientists need to decide which parts of their applications should be executed in cloud computing systems in order to balance the trade-o_ between cost, time and resource requirements. In this paper, we present a service for estimating, monitoring and analyzing costs associated with scientific applications in the cloud. Cost models associated with deferent application execution models are proposed and these cost models can be composed to determine costs of deferent scenarios. We present techniques to estimate costs for service dependency and to monitor costs associated with typical scientific applications. Experiments with real-world applications are performed to illustrate the usefulness of our techniques. Our service could eventually be integrated into cloud resource management and execution services to support on-the-fly resource scheduling. Recently, cloud computing has been considered as an emerging model which aims at allowing customers to utilize computational resources and software hosted by service providers [1, 2, 3], thus shifting the complex and tedious resource and software management tasks typically done by the customers to the service providers. Cloud computing promises to eliminate obstacles due to the management of IT resources and to reduce the cost on infrastructure investments. As a result, besides business customers, cloud computing is also attractive to many scientists from small research groups in the computational science and engineering (CSE) field. However, still there are many unclear questions about how cloud computing can help CSE scientists [3, 1]. Among them, the need to determine the actual cost when using cloud computing is evident. In general, the cost determination is necessary for investigating the return of investment, and, in particular, it is needed to decide when and under which forms cloud computing o_ers can be used. Let us consider this particular point in the interest of small CSE research groups which have limited resources or limited access to their shared computational resources and storages. These groups cannot entirely rely on that resources and storages as well as on cloud computing o_ers due to several reasons. Scientists of these groups need a quick assertion on the cost of executing their applications in the cloud. They want to evaluate if it makes sense to run a particular application or parts of the application using cloud computing, if cloud resources should be used in a regular or occasional basis, and if all resources of need are fully or partially based on clouds. Using information provided by the vendor, it is very midcult to calculate the cost of an application because scientists need to map the computation and data transfer requirements associated with their CSE applications to primitive prices of CPUs, storages and network transfer. Scientists expect to have cost models associated with application models, e.g., Open, MPI, and workflows. Furthermore, a scientific experiment might include deferent parts, each may have a deferent application model and might or might not be executed in the cloud. Thus, it is interesting to have compostable cost models in order to decide which parts of the experiment should be executed by using cloud resources. In this paper, we present a service for estimating and monitoring costs associated with CSE applications that is, in particular, suitable for scientists from small CSE groups. These scientists have some particular constraints that require them to use deferent resources in deferent ways based on fully on-premise, partially cloud, and fully cloud resources. We present compostable cost models. Based on these models, we develop a service which supports both cost estimation and real-time monitoring of costs for CSE applications in the cloud. The rest of this paper is organized as follows. We present requirements for application cost models in Section 2. The compostable cost models are described in Section 3. Our cost estimation, monitoring and analysis service is presented in Section 4. Experiments are described in Section 5. Section 6 discusses the related work.

B. The Cost of Doing Science on the Cloud: The Montage Example

Utility grids such as the Amazon EC2 cloud and Amazon S3 offer computational and storage resources that can be used on-demand for a fee by

compute and data-intensive applications. The cost of running an application on such a cloud depends on the compute, storage and communication resources it will provision and consume. Different execution plans of the same application may result in significantly different costs. Using the Amazon cloud fee structure and a real-life astronomy application, we study via simulation the cost performance tradeoffs of different execution and resource provisioning plans. We also study these trade-offs in the context of the storage and communication fees of Amazon S3 when used for longterm application data archival. Our results show that by provisioning the right amount of storage and compute resources, cost can be significantly reduced with no significant impact on application performance Over the years the research community has developed a wide spectrum of funding and usage models to address the ever growing need for processing, storage, and network resources. From locally owned clusters to national centers and from campus grids to national grids, researchers combine campus and federal funding with competitive and opportunistic compute time allocations to support their science. In some cases, research projects are using their own clusters or pool their resources with other communities (for example in the Open Science Grid (OSG) [1]), or they apply for compute cycles on the national and international cyber infrastructure resources such as those of the Turgid [2] or the EGEE project [3]. Each of these solutions requires a different level of financial commitment and delivers different levels of service. When a project purchases a cluster, this cluster may be expensive but it is fully dedicated to the needs of the project. When joining the OSG, a project contributes some of their resources to the overall collaboration while being able to tap into the capabilities provided by other members of the community. The resource provider still has control over their own resources and may decide on how to share them with others. Providers can also potentially gain the capacity contributed by other members of the collaboration. This system works on the principle that not all the resources are needed at the same time, and when a project does not need their own resources, these cycles are made available to others in the broader collaboration. Another model of computing is delivered by the Turgid, which is a national-level effort to provide a large-scale computational platform for science. Instead of funding individual clusters for individual science projects, it pools together the financial resources of the National Science Foundation to deliver high-performance computing to a broad range of applications. Research projects can apply for allocations of compute cycles that allow them to execute jobs on particular clusters or across the Turgid resources. However, the quality of service is not routinely guaranteed on the Turgid. Although reservations [4], and "urgent computing" [5] are becoming available, an application may not be able to obtain the necessary resources when they are needed (for example, advance reservations generally require one week advance notice). A new dimension to the research computing landscape is added by the cloud computing business model [6]. Based on the economy of scale and advanced web and networking technologies, cloud operators such as Amazon [7] and Google [8] aim to offer researchers as many resources as they need when they need them for as long as they need them. Cloud providers charge applications for the use of their resources according to a fee structure. In addition to supporting on-demand computing, clouds, which use virtualization technologies, enable applications to set up and deploy a custom virtual environment suitable for a given application. Cloud-based outsourcing of computing may be attractive to science applications because it can potentially lower the costs of purchasing, operating, maintaining, and periodically upgrading a local computing infrastructure. In this paper we ask the question: given the availability of clouds, how can an application use them in a way that strikes the right balance between cost and performance. In particular we examine the cost of running on the cloud in the context of an astronomy application Montage [9], which delivers science-grade mosaics of the sky to the community composed of both professional and amateur astronomers. We want to find out what it would mean for a project such as Montage to rely on the cloud, such as the one provided by Amazon [7] to: 1) handle sporadic overloads of mosaic requests, 2) provide resources for all its computations, and 3) support both computation and long-term data storage. Finally we also ask a domain question: how

much would it costs to compute the mosaic of the entire sky on the cloud. The rest of the paper is organized as follows: Section 2 describes the application that motivated this work, Section 3 describes the Amazon computational model we use for the experiments. Section 4 refines the goals of this work, Sections 5 and 6 give an overview of the simulator used in the studies and present the results. Related work is shown in Section 7. Section 8 concludes the paper. the graph and the edges represent the data dependencies between the tasks in the workflow. The numbers in the vertices represent the level of the task in the workflow. The tasks that are not data dependent on other tasks are designed level one. The level of any other task is one plus the maximum level of any of its parent tasks. For the montage workflow, all the tasks at a particular level are invocations of the same routine operating on different input data. For example, the tasks at level one are invocations of the routine mProject which reprojects an input image to the scale defined in an input file called the template header file. This header file is used by all the tasks at level one. The re projected images produced by the tasks at level one are further processed by the tasks at level two as indicated by the edges between the tasks at the two levels. Montage is a data-intensive application. The input images, the intermediate files produced during the execution of the workflow and the output mosaic are of considerable size and require significant storage resources. The tasks on the other hand have a small runtime of at most a few minutes. Section 6.3 quantifies the communication to computation ratio of the montage workflow. As a result of these characteristics, it is desirable to run the montage application in a resource rich environment where the availability of storage resources can be assured.

C. *Providing Database as a Service*

In this paper, we explore a new paradigm for data management in which a third party service provider hosts ”database as a service” providing its customers seamless mechanisms to create, store, and access their databases at the host site. Such a model alleviates the need for organizations to purchase expensive hardware and software, deal with software upgrades, and hire professionals for administrative and maintenance tasks which are taken over by the service provider. We have developed and deployed a database service on the Internet, called NetDB2, which is in constant use. In a sense, data management model supported by NetDB2 provides an effective mechanism for organizations to purchase data management as a service, thereby freeing them to concentrate on their core businesses. Among the primary challenges introduced by ”database as a service” are additional overhead of remote access to data, an infrastructure to guarantee data privacy, and user interface design for such a service. These issues are investigated in the study. We identify data privacy as a particularly vital problem and propose alternative solutions based on data encryption. This paper is meant as a challenges paper for the database community to explore a rich set of research issues that arise in developing such a service. Advances in the networking technologies have triggered one of the key industry responses, the ”software as a service” initiative, also referred to as the application service provider (ASP) model. In this paper, we explore the ”*database as service*” paradigm and the challenges introduced by that. Today, efficient data processing is a fundamental and vital issue for almost every scientific, academic, or business organization. Therefore the organizations end up installing and managing database management systems to satisfy different data processing needs. Although it is possible to purchase the necessary hardware, deploy database products, establish network connectivity, and hire the professional people who run the system, as a traditional solution, this solution has been getting increasingly expensive and impractical as the database systems and problems become larger and more complicated. As it is described above, the traditional solution entails different costs. It might be arguable that hardware, software, and network costs are decreasing constantly. People costs, however, generally, do not decrease. In the future, it is likely that computing solution costs will be dominated by people costs [13]. There is need for database backup, database restore, and database reorganization to reclaim space or to restore preferable arrangement of data. Migration from one database version to the next, without

impacting solution availability, is an art still in its infancy [5]. Parts of a database solution, if not the entire solution usually become unavailable during version change. An organization that provides database service has an opportunity to do these tasks and offer a value proposition provided it is efficient. The new paradigm challenges the traditional model of data management followed by current organizations. Database service provider provides seamless mechanisms for organizations to create, store, and access their databases. Moreover, the entire responsibility of database management, i.e., database backup, administration, restoration, database reorganization to reclaim space or to restore preferable arrangement of data, migration from one database version to the next without impacting availability will befall such an organization. Users wishing to access data will now access it using the hardware and software at the service provider instead of their own organization's computing infrastructure. The application would not be impacted by outages due to software, hardware and network ing changes or failures at the database service provider's site. This would alleviate the problem of purchasing, installing, maintaining and updating the software and administrating the system. Instead of doing these, the organization will only use the ready system maintained by the service provider for its database needs. The technological aspects of developing database as a service lead to new research challenges. First and foremost is the issue of *data privacy*. In the database service provider model, user data needs to reside on the premises of the database service provider. Most corporations view their data as a very valuable asset. The service provider would need to provide sufficient security measures to guard the data privacy. We propose data encryption as the solution to this problem. Detailed investigation of this solution is presented in Section 5. Second key challenge is that of performance. Since the interaction between the users and the database service provider takes place in a different medium, the network, than it does in traditional databases, there are potential overheads introduced by this architecture. Therefore the sources of performance degradation and its significance should be determined. Another challenge facing the database service provider model is that of an appropriate *user interface*. Clearly, the interface must be easy to use; yet it needs to be powerful enough to allow ease in building applications. We have developed and deployed a database service on the Internet, called NetDB2, an experimental networkbasedapplication service provider (ASP) system. It has been operational over a year and used by number of universities to help teaching database courses at different locations. NetDB2 provides database services including tools for application development, creating and loading tables, and performing queries and transactions to the users over the Internet.

D. Hierarchical Attribute-Based Encryption for Fine-Grained Access Control in Cloud Storage Services

Cloud computing, as an emerging computing paradigm, enables users to remotely store their data into a cloud so as to enjoy scalable services on-demand. Especially for small and medium-sized enterprises with limited budgets, they can achieve cost savings and productivity enhancements by using cloud-based services to manage projects, to make collaborations, and the like. However, allowing cloud service providers (CSPs), which are not in the same trusted domains as enterprise users, to take care of confidential data, may raise potential security and privacy issues. To keep the sensitive user data confidential against entrusted CSPs, a natural way is to apply cryptographic approaches, by disclosing decryption keys only to authorized users. However, when enterprise users outsource confidential data for sharing on cloud servers, the adopted encryption system should not only support fine-grained access control, but also provide high performance, full delegation, and scalability, so as to best serve the needs of accessing data anytime and anywhere, delegating within enterprises, and achieving a dynamic set of users. In this paper, we propose a scheme to help enterprises to efficiently share confidential data on cloud servers. We achieve this goal by first combining the hierarchical identity-based encryption (HIBE) system and the cipher text-policy attribute-based encryption (CP-ABE) system, and then making a performance-expressivity tradeoff, finally applying proxy re-encryption and lazy re-encryption to our scheme. With the emergence of sharing confidential corporate data on cloud servers, it is imperative to adopt an efficient encrypt

tion system with a fine-grained access control to encrypt outsourced data. Ciphertext-policy attribute-based encryption (CP-ABE), as one of the most promising encryption systems in this field, allows the encryption of data by specifying an access control policy over attributes, so that only users with a set of attributes satisfying this policy can decrypt the corresponding data. However, a CP-ABE system may not work well when enterprise users outsource their data for sharing on cloud servers, due to the following reasons: First, one of the biggest merits of cloud computing is that users can access data stored in the cloud anytime and anywhere using any device, such as thin clients with limitedbandwidth, CPU, and memory capabilities. Therefore, the encryption system should provide high performance. Second, in the case of a large-scale industry, a delegation mechanism in the generation of keys inside an enterprise is needed.

Although some CP-ABE schemes support delegation between users, which enables a user to generate attribute secret keys containing a subset of his own attribute secret keys for other users, we hope to achieve a full delegation, that is, a delegation mechanism between attribute authorities (AAs), which independently make decisions on the structure and semantics of their attributes. Third, in case of a large-scale industry with a high turnover rate, a scalable revocation mechanism is a must. The existing CP-ABE schemes usually demand users to heavily depend on AAs and maintain a large amount of secret keys storage, which lacks flexibility and scalability. Motivation**.** Our main design goal is to help the enterprise users to efficiently share confidential data on cloud servers. Specifically, we want to make our scheme more applicable in cloud computing by simultaneously achieving fine-grained access control, high performance, practicability, and scalability. Our Contribution. In this paper, we first propose a hierarchical attribute-based encryption (HABE) model by combining a HIBE system and a CP-ABE system, to provide fine-grained access control and full delegation. Based on the HABE model, we construct a HABE scheme by making a performance-expressivity tradeoff, to achieve high performance. Finally, we propose a scalable revocation scheme by delegating to the CSP most of the computing tasks in revocation, to achieve a dynamic set of users efficiently.

E. Fully Homomorphism Encryption Using Ideal Lattices

We propose a fully homomorphism encryption scheme – i.e., a scheme that allows one to evaluate circuits over encrypted data without being able to decrypt. Our solution comes in three steps. First, we provide a general result – that, to construct an encryption scheme that permits evaluation of arbitrary circuits, it suffices to construct an encryption scheme that can evaluate (slightly augmented versions of) its own decryption circuit; we call a scheme that can evaluate its (augmented) decryption circuit boots trappable. Next, we describe a public key encryption scheme using ideal lattices that is almost boot strippable. Lattice-based cryptosystems typically have decryption algorithms with low circuit complexity, often dominated by an inner product computation that is in NC1. Also, ideal lattices provide both additive and multiplicative homeomorphisms (modulo a public-key ideal in a polynomial ring that is represented as a lattice), as needed to evaluate general circuits. Unfortunately, our initial scheme is not quite bootstrap- pable – i.e., the depth that the scheme can correctly evalu- ate can be logarithmic in the lattice dimension, just like the depth of the decryption circuit, but the latter is greater than the former. In the final step, we show how to modify the scheme to reduce the depth of the decryption circuit, and thereby obtain a boot strippable encryption scheme, with- out reducing the depth that the scheme can evaluate. Ab- stractly, we accomplish this by enabling the encrypter to start the decryption process, leaving less work for the de-crypter, much like the server leaves less work for the de- creeper in a server-aided cryptosystem. We propose a solution to the old open problem of con-structing a fully homomorphism encryption scheme.

This no- tion, originally called a privacy homomorphism, Here, we focus on constructing a scheme that is semanti-cally secure against chosen plaintext attacks (or just “semantically secure”). Unfortunately a scheme that has nontriv- ial homeomorphisms’ cannot be semantically secure against adaptive chosen cipher text attacks (CCA2), since it is mal- leable. There are relaxed notions of CCA2 security [3, 16, 52], but they do not apply to a

fully homomorphism scheme. However, constructing a CCA1-secure fully homomorphism encryption scheme is an interesting open problem. We construct a fully homomorphism encryption scheme us- ing ideal lattices. The result can roughly be broken down into three steps: a general "bootstrapping" result, an "initialconstruction" using ideal lattices, and a technique to "squash the decryption circuit" to permit bootstrapping. Our research began with the second step: a PKE scheme E1 described in Section 3 that uses ideal lattices and is homo- morphed for shallow circuits. A cipher text has the form v + x where v is in the ideal lattice and x is an "error" or "offset" vector that encodes the plaintext _. Interpret- ing cipher text vectors as coefficient vectors of elements in a polynomial ring Z[x]/f(x), we add and multiply cipher texts using ring operations – 1 + 2 or 1 × 2 – and induce addition and multiplication of the underlying plain- texts. By itself, this scheme improves upon prior work. It compares favorably to Boneh-Goh-Nissim [11]; it is homo-morphic for circuits with greater multiplicative depth while allowing essentially unlimited additions. The security of E1 is based on a natural decisional version of the closest vector problem for ideal lattices for ideals in a fixed ring. E1 is homomorphism only for shallow circuits because the "error" vector grows with addition and (especially) multi-plication operations; eventually, it becomes so long that it causes a decryption error. It is natural to ask: can we "re- fresh" a cipher text whose error vector is almost too long to obtain a new cipher text whose error vector is shorter? Ob- viously, we could refresh a cipher text if we could completely Homomorphism encryption schemes that are not semantic- ally secure, like basic RSA, may also have stronger attacks on their one-wayness. Boneh and Lipton [13] proved that any algebraic privacy homomorphism over a ring Zn can be broken in sub-exponential time under a (reasonable) num-ber theoretic assumption, if the scheme is deterministic or otherwise offers an equality oracle. See also [35] and [18]. Goldwasser and Micali [23] proposed the first semantic- cally secure homomorphism encryption scheme (and also in- troduced the notion of semantic security).

Their scheme is "additively homomorphism" over Z2; in our terminology, its set CE of permitted circuits contains circuits with XOR gates. Other additively homomorphism encryption schemes with proofs of semantic security are Benaloh [8], Naccache- Stern [42], Okamoto-Uchiyama [46], Paillier [47], and Damgard- Jurik [19]. Some additively homomorphism encryption schemes use lattices or linear codes [22, 50, 27, 36, 37, 4]. ElGamal [20] is multiplicatively homomorphic. Semantically secure schemes that allow both addition and multiplication include Boneh-Goh-Nissim [11] (quadratic for- mulas) and "Polly Cracker" by Fellows and Koblitz [21, 29, 30, 31] (arbitrary circuits but with exponential ciphertext-size blow-up). Sanders, Young and Yung [56, 7] (SYY) use circuit-private additively homomorphic encryption to con- struct a circuit-private scheme that can handle NC1 circuits. Ishai and Paskin [26] do this for branching programs, which covers NC1 circuits (by Barrington [6]), and ciphertexts in their scheme are much shorter – proportional to the length of the branching program rather than its size, though the computation is proportional to the size.

IV. System Analysis

The Systems Development Life Cycle (SDLC), or Software Development Life Cycle in systems engineering, information systems and software engineering, is the process of creating or altering systems, and the models and methodologies that people use to develop these systems. In software engineering the SDLC concept underpins many kinds of software development methodologies. These methodologies form the framework for planning and controlling the creation of an information system the software development process.

A. Software Model or Architecture Analysis

Structured project management techniques (such as an SDLC) enhance management's control over projects by dividing complex tasks into manageable sections. A software life cycle model is either a descriptive or prescriptive characterization of how software is or should be developed. But none of the SDLC models discuss the key issues like Change management, Incident management and Release management processes within the SDLC process, but, it is addressed in the overall project management. In the proposed hypothetical model, the concept of user-developer interaction in the

conventional SDLC model has been converted into a three dimensional model which comprises of the user, owner and the developer. In the proposed hypothetical model, the concept of user-developer interaction in the conventional SDLC model has been converted into a three dimensional model which comprises of the user, owner and the developer. The ―one size fits all‖ approach to applying SDLC methodologies is no longer appropriate. We have made an attempt to address the above mentioned defects by using a new hypothetical model for SDLC described elsewhere. The drawback of addressing these management processes under the overall project management is missing of key technical issues pertaining to software development process that is, these issues are talked in the project management at the surface level but not at the ground level.

B. What is SDLC

A software cycle deals with various parts and phases from planning to testing and deploying software. All these activities are carried out in different ways, as per the needs. Each way is known as a Software Development Lifecycle Model (SDLC). A software life cycle model is either a descriptive or prescriptive characterization of how software is or should be developed. A descriptive model describes the history of how a particular software system was developed. Descriptive models may be used as the basis for understanding and improving software development processes or for building empirically grounded prescriptive models.

SDLC models * The Linear model (Waterfall) - Separate and distinct phases of specification and development. - All activities in linear fashion. - Next phase starts only when first one is complete. * Evolutionary development - Specification and development are interleaved (Spiral, incremental, prototype based, Rapid Application development). - Incremental Model (Waterfall in iteration), - RAD(Rapid Application Development) - Focus is on developing quality product in less time, - Spiral Model - We start from smaller module and keeps on building it like a spiral. It is also called Component based development. * Formal systems development - A mathematical system model is formally transformed to an implementation. * Agile Methods. - Inducing flexibility into development. * Reuse-based development - The system is assembled from existing components.

C. Software Requirements Specification

i. Software Requirements:

Language	:	JDK (1.7.0)
Frontend	:	JSP, Servlets
Backend	:	Oracle10g
IDE	:	my eclipse 8.6
Operating System	:	windows XP
Server	:	tomcat

ii. Hardware Requirements:

Processor	:	Pentium IV
Hard Disk	:	80GB
RAM	:	2GB

D. The General Model :

Software life cycle models describe phases of the software cycle and the order in which those phases are executed. There are tons of models, and many companies adopt their own, but all have very similar patterns. Each phase produces deliverables required by the next phase in the life cycle. Requirements are translated into design. Code is produced during implementation that is driven by the design. Testing verifies the deliverable of the implementation phase against requirements.

i. SDLC Methodology:

Spiral Model: The spiral model is similar to the incremental model, with more emphases placed on risk analysis. The spiral model has four phases: Planning, Risk Analysis, Engineering and Evaluation. A\ software project repeatedly passes through these phases in iterations (called Spirals in this model). The baseline spiral, starting in the planning phase, requirements is gathered and risk is assessed. Each subsequent spirals builds on the baseline spiral. Requirements are gathered during the planning phase. In the risk analysis phase, a process is undertaken to identify risk and alternate solutions. A prototype is produced at the end of the risk analysis phase. Software is produced in the

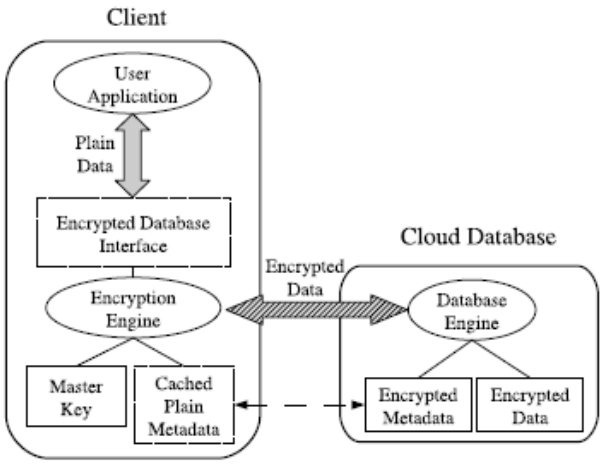
engineering phase, along with testing at the end of the phase. The evaluation phase allows the customer to evaluate the output of the project to date before the project continues to the next spiral. In the spiral model, the angular component represents progress, and the radius of the spiral represents cost. Spiral Life Cycle Model.

This document play a vital role in the development of life cycle (SDLC) as it describes the complete requirement of the system. It means for use by developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

SPIRAL MODEL was defined by Barry Boehm in his 1988 article, "A spiral Model of Software Development and Enhancement. This model was not the first model to discuss iterative development, but it was the first model to explain why the iteration models.

As originally envisioned, the iterations were typically 6 months to 2 years long. Each phase starts with a design goal and ends with a client reviewing the progress thus far. Analysis and engineering efforts are applied at each phase of the project, with an eye toward the end goal of the project.

V. ARCHITECTURE



Modules description:

1. System Model
2. Adaptive Encryption Scheme
3. Cost estimation of cloud database services
4. Performance Evolution

System Model:

The proposed system supports adaptive encryption for public cloud database services, where distributed and concurrent clients can issue direct SQL operations. By avoiding an architecture based on intermediate servers between the clients and the cloud database, the proposed solution guarantees the same level of scalability and availability of the cloud service. A scheme of the proposed architecture where each client executes an encryption engine that manages encryption operations. This software module is accessed by external user applications through the encrypted database interface. The proposed architecture manages five types of information:

1. Plain data represent the tenant information
2. Encrypted data are the encrypted version of the plain data, and are stored in the cloud database;
3. Plain metadata represent the additional information that is necessary to execute SQL operations on encrypted data
4. Encrypted metadata are the encrypted version of the plain metadata, and are stored in the cloud database;
5. Master key is the encryption key of the encrypted metadata, and is known by legitimate clients.

Adaptive Encryption Scheme:

We consider SQL-aware encryption algorithms that guarantee data confidentiality and allow the cloud database engine to execute SQL operations over encrypted data. As each algorithm supports a specific subset of SQL operators, we refer to the following encryption schemes.

1. Random (Rand): it is the most secure encryption because it does not reveal any information about the original plain value (IND-CPA) [20], [21]. It does not support any SQL operator, and it is used only for data retrieval.
2. Deterministic (Det): it deterministically encrypts data, so that equality of plaintext

data is preserved. It supports the equality operator.

3. Order Preserving Encryption (Ope) [12]: it preserves in the encrypted values the numerical order of the original unencrypted data. It supports the comparison SQL operators (i.e., =; <;<_; >;>_).
4. Homomorphic Sum (Sum) [13]: it is homomorphic with respect to the sum operation, so that the multiplication of encrypted integers is equal to the sum of plaintext integers. It supports the sum operator between integer values.
5. Search: it supports equality check on full strings (i.e., the LIKE operator).
6. Plain: it does not encrypt data, but it is useful to support all SQL operators on non confidential data.

Cost estimation of cloud database services:

We consider a tenant that is interested in estimating the cost of porting his database to a cloud platform. This porting is a strategic decision that must evaluate confidentiality issues and related costs over a medium-long term. For these reasons, we propose a model that includes the overhead of encryption schemes and the variability of database workload and cloud prices. The proposed model is general enough to be applied to the most popular cloud database services, such as Amazon Relational Database Service, Enterprise DB, Windows Azure SQL Database, and Rack space Cloud Database.

Performance Evolution:

This section aims to verify whether the overheads of adaptive encryption represent an acceptable compromise between performance and data confidentiality for the tenants of cloud database services. To this purpose, we design a suite of performance tests that allow us to evaluate the impact of encryption and adaptive encryption on response times and throughput for different network latencies and for increasing numbers of concurrent clients.

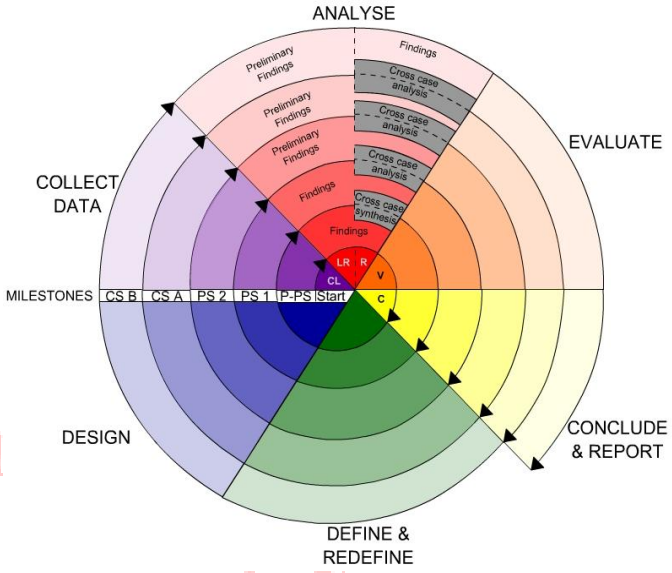


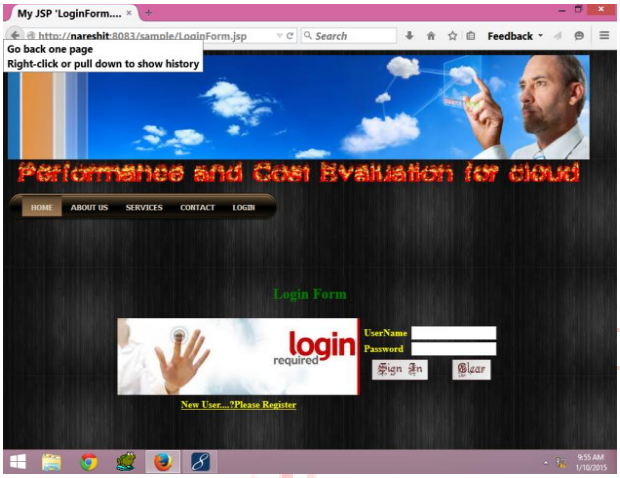
Fig -Spiral Model

CONCLUSION

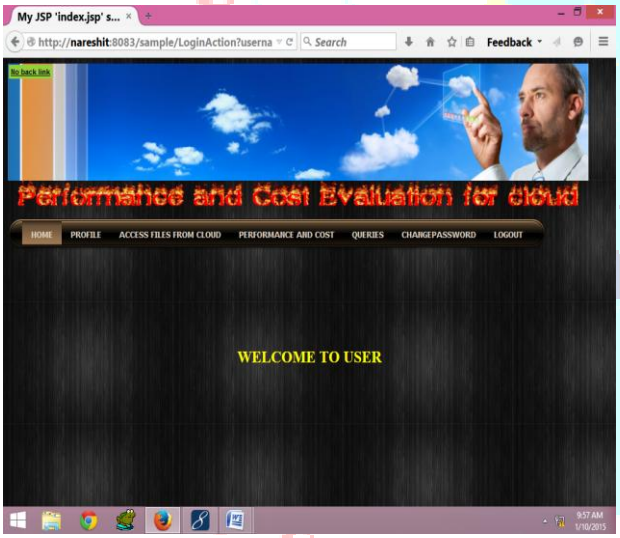
There are two main tenant concerns that may prevent the adoption of the cloud as the fifth utility: data confidentiality and costs. This paper addresses both issues in the case of cloud database services. These applications have not yet received adequate attention by the academic literature, but they are of utmost importance if we consider that almost all important services are based on one or multiple databases. We address the data confidentiality concerns by proposing a novel cloud database architecture that uses adaptive encryption techniques with no intermediate servers. This scheme provides tenants with the best level of confidentiality for any database workload that is likely to change in a medium-term period. We investigate the feasibility and performance of the proposed architecture through a large set of experiments based on a software prototype subject to the TPC-C standard benchmark. Our results demonstrate that the network latencies that are typical of cloud database environments hide most overheads related to static and adaptive encryption.

Result Analysis

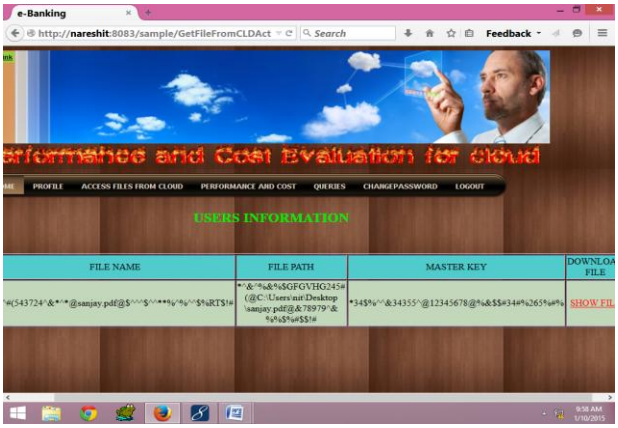
1. Login Screen



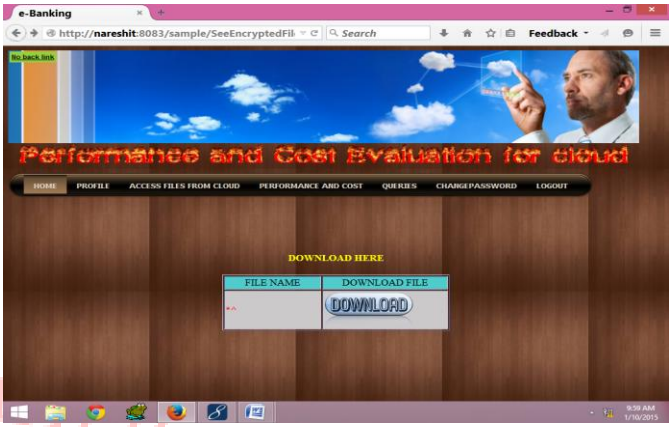
User homepage



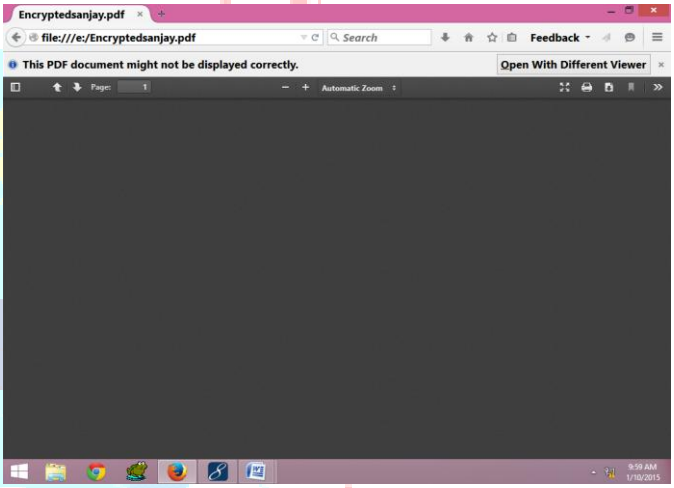
Encrypted metadata



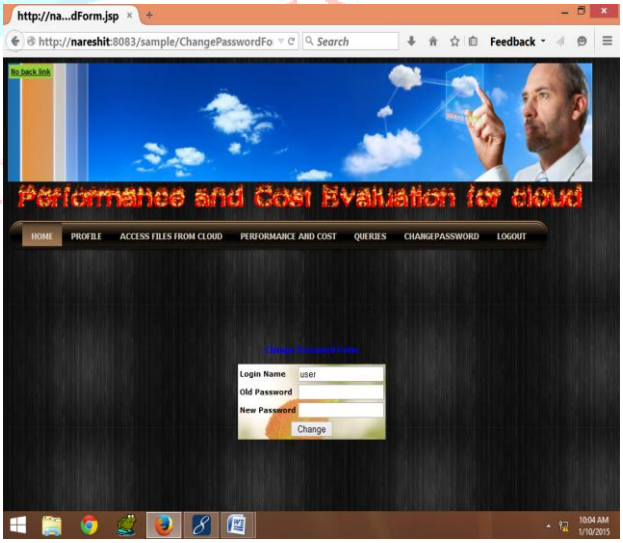
Download encrypted file



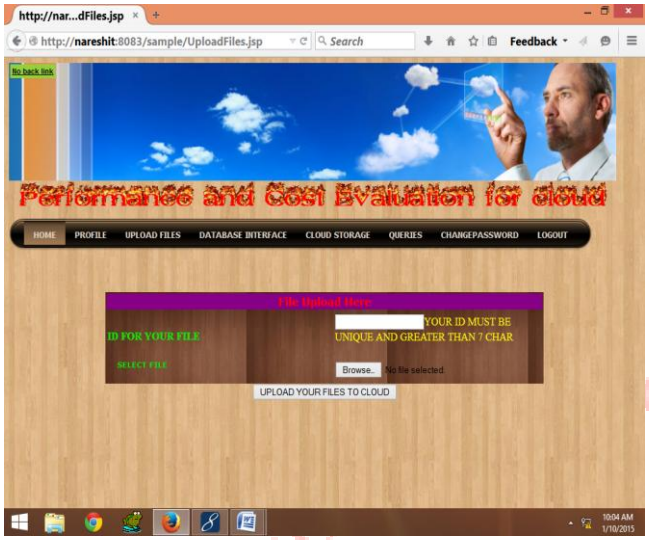
Encrypted pdf file



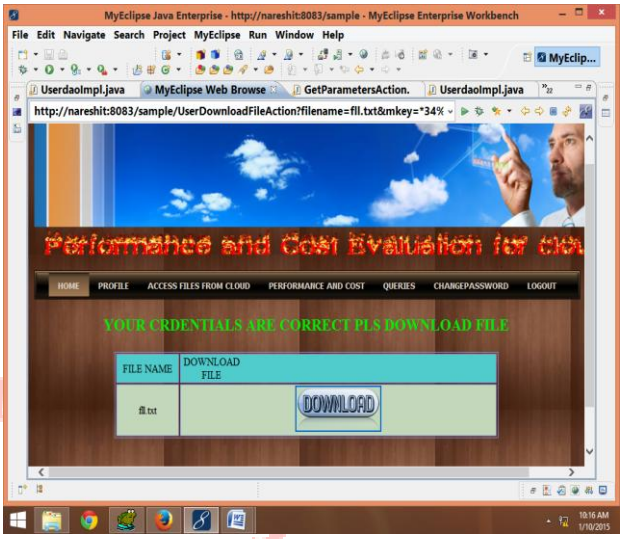
Change password



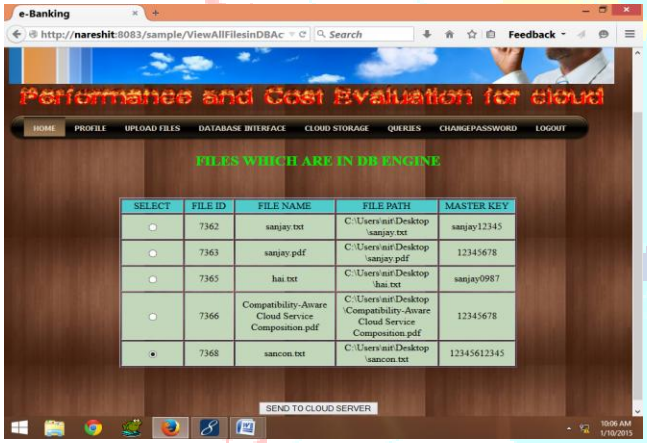
Admin upload file



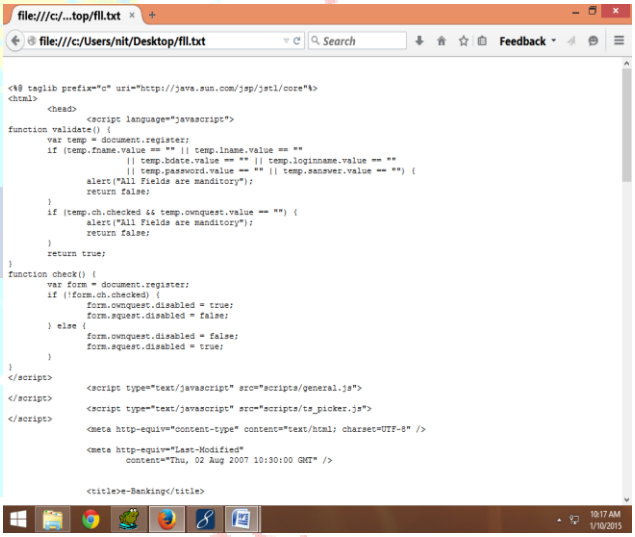
Adaption layer removal



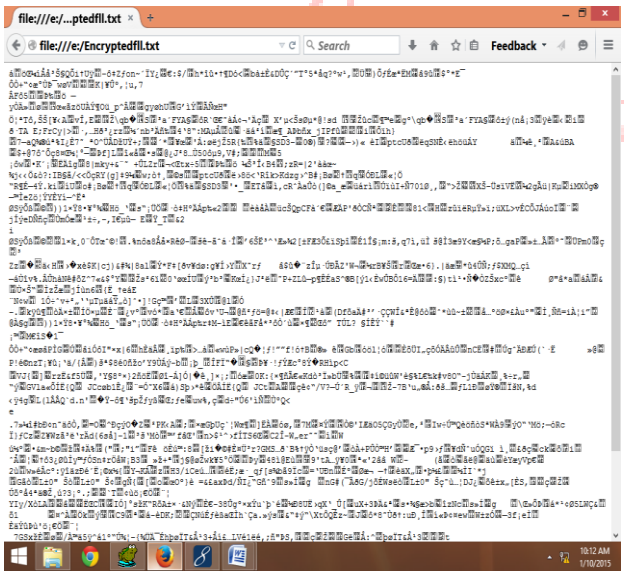
View all files in DB



File after adaption layer removal



Encrypted file in cloud



REFERENCES

[1] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, “Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility,” Future Generation Comput. Syst., vol. 25, no. 6, pp. 599–616, 2009.

[2] T. Mather, S. Kumaraswamy, and S. Latif, Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance. Sebastopol, CA, USA: O’Reilly Media, Inc., 2009.

[3] H.-L. Truong and S. Dustdar, “Composable cost estimation and monitoring for computational applications in cloud computing environments,” Procedia Comput. Sci., vol. 1, no. 1, pp. 2175–2184, 2010.

[4] E. Deelman, G. Singh, M. Livny, B. Berriman, and J. Good, “The cost of doing science on the cloud: The montage

example," in Proc. ACM/IEEE Conf. Supercomputing, 2008, pp. 1–12.

[5] H. Hacig€um€u¸s, B. Iyer, and S. Mehrotra, "Providing database as a service," in Proc. 18th IEEE Int. Conf. Data Eng., Feb. 2002, pp. 29–38.

[6] G. Wang, Q. Liu, and J. Wu, "Hierarchical attribute-based encryption for fine-grained access control in cloud storage services," in Proc. 17th ACM Conf. Comput. Commun. Security, 2010, pp. 735–737.

[7] Google. (2014, Mar.). Google Cloud Platform Storage with serverside encryption [Online]. Available: blogspot.it/2013/08/google-cloud-storage-now provides.html.

[8] H. Hacig€um€u¸s , B. Iyer, C. Li, and S. Mehrotra, "Executing SQL over encrypted data in the database-service-provider model," in Proc. ACMSIGMODInt'l Conf. Manage. Data, Jun. 2002, pp. 216–227.

[9] L. Ferretti, M. Colajanni, and M. Marchetti, "Distributed, concurrent, and independent access to encrypted cloud databases," IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 2, pp. 437–446, Feb. 2014. TABLE 6 Costs of the Cloud Database Service during the Three Years Period in STATIC and DYNAMIC Scenarios 154 IEEE TRANSACTIONS ON CLOUD COMPUTING, VOL. 2, NO. 2, APRIL-JUNE 2014

[10] R. A. Popa, C. M. S. Redfield, N. Zeldovich, and H. Balakrishnan, "CryptDB: Protecting confidentiality with encrypted query processing," in Proc. 23rd ACM Symp. Operating Systems Principles, Oct. 2011, pp. 85–100.

Author Profiles:

Ramana Reddy M.Tech student in CSE with specialization (computer science engineering) from Maheshwara Engineering college. His areas of research interests include Networks, Analysis of Algorithm, Compiler and Language Processing.